

基于GPP-SDR的HSUPA接收机软件设计及SSE指令优化*

王俊华 厦门大学信息科学与技术学院
齐 心 清华大学信息科学与技术国家实验室
黄联芬 厦门大学信息科学与技术学院

【摘要】文章首先介绍了TD-HSUPA信道，然后对TD-HSUPA接收机的软件设计及SSE优化进行重点分析，并给出了优化前后的运行时间对比。通过对关键算法模块采用SSE指令进行优化，可极大地减少运行时间，提高软件运行效率。

【关键词】TD-SCDMA HSUPA GPP-SDR SSE FFT

1 引言

TD-HSUPA标准化工作是在3GPP的Release 7中引入的无线侧上行链路增强技术，它采用的关键技术有高速上行共享信道、混合自动重传（HARQ）、自适应调制编码（AMC）、16QAM技术和快速调度等技术^[1]。

作为3G网络建设中的一种重要技术，TD-HSUPA可以在网络结构不变的情况下，极大提高上行业务速率。但因其基站端接收处理的复杂度，协议版本更新快，灵活性和可重构性差，因此软件无线电（SDR，Software Definition Radio）技术被工业界广泛使用。当前，SDR平台主要分为：现场可编程门阵列（FPGA）、数字信号处理器（DSP）和通用处理器（GPP）等。FPGA、DSP虽然处理能力比较强，但是需要巨大的投入^[2]。

通过GPP-SDR平台，设计和实现TD-HSUPA的基站侧

接收机。在接收端处理中，有些模块的计算量比较大，为了提高软件运行效率，在实现中使用了SSE（Streaming SIMD Extensions）指令。SSE是Intel发布的对单指令多操作（SIMD）指令流的扩展指令，其优势包括计算、处理并行度高，提高程序运行效率更高。

2 TD-HSUPA信道介绍

在TD-SCDMA中^[3]，一个无线帧的长度为10ms，分为两个5ms的子帧，每个子帧包含有上行同步时隙、下行同步时隙、保护时隙和7个固定长度的业务时隙。其中，TS0固定用于下行方向，TS1固定用于上行方向，其余时隙可以灵活配置，如图1所示：

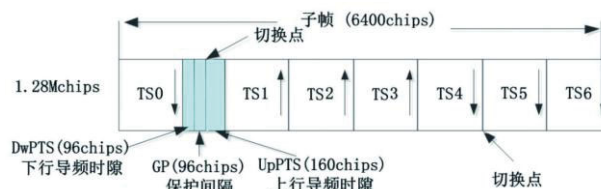


图1 TD-SCDMA系统的子帧结构

*基金项目：IBM开放合作研究项目IBM SUR/OCR（20122X03003007-004）

收稿日期：2011-12-23

TDD模式下的物理信道是一个突发信道，由864个码片组成。其中，Midamble部分占144个码片，数据部分占704个码片，保护间隔GP部分占16个码片，如图2所示：

数据	Midamble	数据	GP
352chips	144chips	352chips	16chips

图2 TD-SCDMA 的突发结构

在3GPP的Release 7中，TD-HSUPA引入了一条增强专用信道（E-DCH）和五条物理信道^[4]（包括三条上行物理信道E-RUCCH、E-PUCH、E-UCCH以及两条下行物理信道E-AGCH、E-HICH）。下面是对各个信道的具体介绍：

E-DCH是一条新增的上行传输信道，用于承载HSUPA上行数据，在一个5ms的TTI中承载一个MAC-e传输块，并且映射到物理信道E-PUCH的数据部分。

E-RUCCH是一条新增的上行物理信道，用于发送在UE上辅助Node-B调度的相关信息，并映射到随机接入物理资源中。

E-AGCH是一条新增的传输Node-B下发的资源授权信息的下行物理信道。在物理层中，E-AGCH是两条物理信道的组合，分别为E-AGCH1和E-AGCH2。

E-UCCH是一条新增的用于承载E-DCH上行控制信息的上行物理信道，与E-DCH共同映射到E-PUCH中。

E-PUCH是一条新增的用于承载E-DCH上行数据和E-UCCH控制信息的上行物理信道。E-PUCH的突发格式有两种：包含E-UCCH、TPC和不包含E-UCCH、TPC，是否带有E-UCCH、TPC由高层决定。

E-HICH是一条新增的传输Node-B中HARQ实体的确认信息的下行物理信道，它是由一条 $SF=16$ 的下行物理信道和一个签名序列所定义。

TD-HSUPA的调度过程如图3所示。

(1) UE通过E-RUCCH发起调度请求，其调度相关信息包含小区之间的路径损耗信息、可使用的功率信息、缓存占用信息以及UE的标识等。

(2) Node-B接收到E-RUCCH的调度请求后，判断

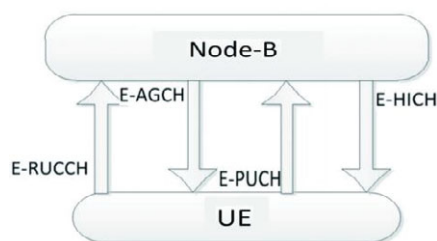


图3 TD-HSUPA调度过程

是否允许该UE发送上行增强数据，如果允许，则通过E-AGCH发送接入允许信息给UE。

(3) UE收到E-AGCH的信息，解出自己所需要的信息后，根据分配的资源 and 功率在E-DCH上选择合适的速率开始传输数据。

(4) Node-B接收到E-DCH信息，根据解调后的数据正确与否，在对应的E-HICH信道上反馈ACK/NACK信息，UE根据反馈信息去判断是否需要重传。

3 接收机的软件设计

本文基于GPP-SDR平台对TD-HSUPA的接收端进行了软件设计^[5~7]，模块框图如图4所示。

从图4可以看出，对E-RUCCH/E-AGCH/E-HICH信道而言，由于其扩频因子（ $SF=16$ ），使用Rake接收来消除码间干扰；但是对于E-PUCH信道而言，没有使用扩频（ $SF=1$ ），故采用均衡技术来消除码间干扰。

在整个接收机的软件设计中，为了使传输速率更快，需要对其进行优化，本文只对频域均衡和解调模块进行优化，其余模块作为后续工作。

3.1 频域均衡

在无线信道中，由于存在多径问题，数据的传输会产生符号间干扰（ISI），为了有效地克服ISI，本系统中使用了均衡技术。均衡技术可分为时域均衡和频域均衡，时域均衡技术虽然在通信中得到了广泛的应用，但是其运算复杂度高，故本文使用频域均衡来消除ISI。

频域均衡的基本思想是估计出信道的频率响应（ H_k ），然后在每个子信道上乘以均衡系数（ W_k ）来补偿信道的影响^[8]。在时变信道中， H_k 是通过每一帧数据的

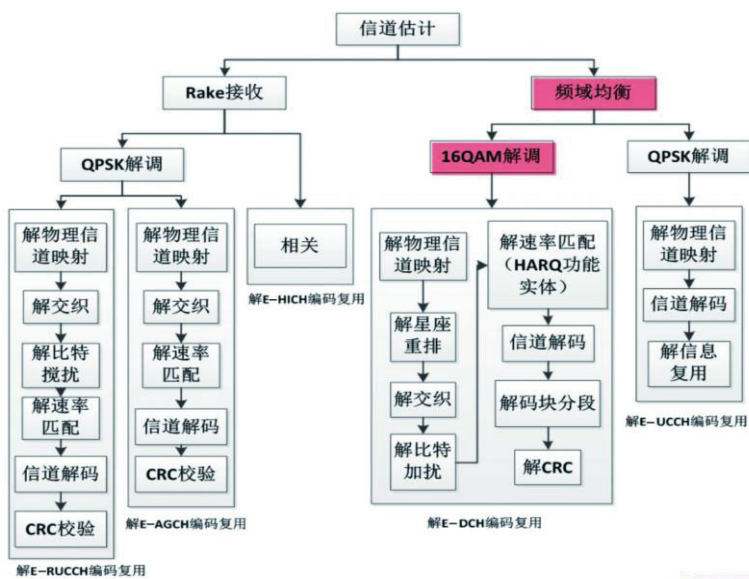


图4 TD-HSUPA接收机的模块设计

导频部分进行信道估计得出的。例如， X_k 为发送的导频信号， Y_k 为接收的导频信号，对二者做 L 点的FFT变换得到 $\text{fft}(X_k)$ 和 $\text{fft}(Y_k)$ ，则 L 点的信道频率响应的估计值 \tilde{H}_L 表示为：

$$\tilde{H}_L = \frac{\text{fft}(x_k)}{\text{fft}(y_k)}, k=1,2,3,\dots,L-1 \quad (1)$$

对 \tilde{H}_L 做 L 点的IFFT变换，得到信道的冲击响应 \tilde{h} 。然后在 \tilde{h} 的末尾补零，使序列补到合适的长度 N ，再对 \tilde{h} 进行 N 点的FFT变换，得到 N 点的信道频率响应的估计值 \tilde{H}_N 。均衡的目标是码间干扰为零，滤波器的均衡系数可以由以下公式得到：

$$W_M = \frac{1}{H_M}, M=1,2,3,\dots,N-1 \quad (2)$$

在实际的通信系统中，可以用 \tilde{H}_N 代替 H 。例如，设 $x(i)$ 为输入信号， W 为抽头系数， $s(i)$ 为输出信号，则由公式(2)可知，整个频域均衡表示为：

$$s(i) = \text{ifft} \{ \text{fft}[x(i)] * W \} \quad (3)$$

对于本系统而言，冲击响应 h 应补齐到1024chips，输入数据 x 也应该补齐到1024chips，通过公式(3)，可得：

$$y = \text{ifft} \left(\frac{\text{fft}(x)}{\text{fft}(h)} \right) \quad (4)$$

在此输入记为 x ，频域均衡输出为 y 。在频域均衡的

过程中，FFT因其复杂的数据运算处理算法制约着频域均衡整体处理时间，可使用具有并行处理能力的SSE指令去减少程序运行时间。

3.2 1024点FFT的SSE优化

FFT并不是一种新的变换形式，它只是DFT的一种快速算法^[9]。主要分为两大类：按时间抽选法和按频率抽选法。本文使用了按时间抽选法，按频率抽选法也类似。

由于1024点的FFT变换的蝶形图太大，这里只介绍8点FFT变换的SSE优化，1024点也类似。如图5所示，为8点FFT变换的蝶形图。

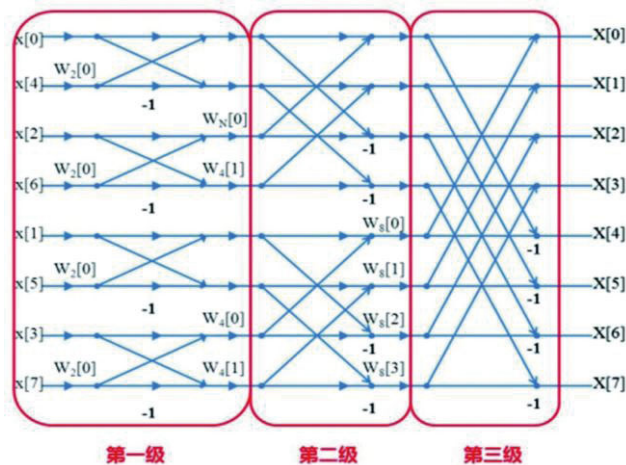


图5 8点FFT蝶形图

由图5可知，输入与输出有一个排序的过程，每进行一级变换都要进行一次奇偶排序，这个过程称为FFT的逆位序。

这里需要输出为正位序，所以需要输入序列为逆位序排列，在逆位序排序时使用了制表的方法，具体是：定义一个数组reverse，把计算好的逆位序数放入数组reverse中，以8点FFT为例，逆位序排序为 $\text{reverse}[8] = \{0, 4, 2, 6, 1, 5, 3, 7\}$ ，假设原始输入数据为 $\{x(0), x(1), x(2), x(3), x(4), x(5), x(6), x(7)\}$ ，那么第一级的输入数据为 $x[\text{reverse}[i]]$ ，变成 $\{x(0), x(4), x(2), x(6), x(1), x(5), x(3), x(7)\}$ ，这样的话就可以通过查找

数组reverse来进行逆位序的排列,从而节省计算逆位序的时间。

在第一级中,每两个输入数据为一组,它们之间有交叉运算。在每组数据交叉运算之前,每个数据都要乘以对应的旋转因子,对应于8点FFT,交叉运算结果为:

$$\begin{cases} X(0) = x(0) + w_2^0 x(4) \\ X(1) = x(0) - w_2^0 x(4) \end{cases}$$

这里 $X(0)$ 、 $X(1)$ 为第一级的输出,同理可以计算出 $\{x(2), x(6)\}$ 、 $\{x(1), x(5)\}$ 和 $\{x(3), x(7)\}$ 这三组数据,旋转因子 W 也使用制表的方法,可节省数据的计算时间。SSE指令每次处理128bits数据,每次只能处理4个float型数据,由于在第一级中逆位序的影响,使相邻的数据不连续排列,因此第一级不能使用SSE指令进行编写。

在第二级中,每4个输入数据为一组。在运算之前,首先要乘以相应的旋转因子 w_4^0 、 w_4^1 ,在8点FFT变换中,数据分为两组,每组4个,假设第二级输出 $Y(i)$,那么可以得到:

$$\begin{cases} Y(0) = X(0) + w_4^0 X(2) \\ Y(1) = X(0) - w_4^1 X(2) \end{cases} \quad \begin{cases} Y(2) = X(1) + w_4^0 X(3) \\ Y(3) = X(1) - w_4^1 X(3) \end{cases}$$

$$\begin{cases} Y(4) = X(4) + w_4^0 X(6) \\ Y(5) = X(4) - w_4^1 X(6) \end{cases} \quad \begin{cases} Y(6) = X(5) + w_4^0 X(7) \\ Y(7) = X(5) - w_4^1 X(7) \end{cases}$$

其中, $X(i)$ 为第一级的输出。可以看出,每组数据之间有交叉运算,这就需要用SSE指令_mm_shuffle_ps进行内部位置调整,然后使用_mm_add_ps、_mm_sub_ps、_mm_mul_ps指令进行相加、减、乘即可。

在第三级中,每4个数据内部没有交叉,可以直接使用SSE指令去处理数据,根据图5得出:

$$\begin{cases} Z(0) = Y(0) + w_8^0 Y(4) \\ Z(1) = Y(0) - w_8^1 Y(4) \end{cases} \quad \begin{cases} Z(2) = Y(1) + w_8^2 Y(5) \\ Z(3) = Y(1) - w_8^3 Y(5) \end{cases}$$

$$\begin{cases} Z(4) = Y(2) + w_8^0 Y(6) \\ Z(5) = Y(2) - w_8^1 Y(6) \end{cases} \quad \begin{cases} Z(6) = Y(3) + w_8^2 Y(7) \\ Z(7) = Y(3) - w_8^3 Y(7) \end{cases}$$

这里 $Z(i)$ 为第三级的输出,在这级中只使用了mm_add_ps、_mm_sub_ps、_mm_mul_ps指令进行相加减,与第二级的优化方式相似。

以上是对8点FFT进行优化,用同样的方法也可以计算出1024点的FFT变换。

3.3 软解调的SSE优化

软解调与硬解调相比能够极大提高译码性能,在进行硬判决译码时,采用码字间汉明距离最大化准则。而对于软判决译码,则采用欧氏距离最大化准则^[10]。

在说明软判决前,先简单介绍下硬判决。对于比特 $b_{l,k}$ (同样适用于比特 $b_{Q,k}$)最佳硬判决的准则如下:

$$\hat{b}_{l,k} = \beta, \text{ 若 } p[b_{l,k} = 1 | r[i]] > p[b_{l,k} = 0 | r[i]], \quad \beta = 0, 1 \quad (5)$$

其中, $r[i]$ 为接收端收到的信号, $r(i) = G_{ch}(i) a(i) + n(i)$,这里 $G_{ch}(i)$ 是随机复信道增益, $n(i)$ 是复高斯随机噪声。假设 $\beta = 1$,那么公式(5)就可以写作:

$$\hat{b}_{l,k} = 1, \text{ 若 } \lg \frac{p[b_{l,k} = 1 | r[i]]}{p[b_{l,k} = 0 | r[i]]} > 0 \quad (6)$$

因此,判决 $\hat{b}_{l,k}$ 的对数似然比被定义为:

$$LLR(b_{l,k}) = \lg \frac{p[b_{l,k} = 1 | r[i]]}{p[b_{l,k} = 0 | r[i]]} = \lg \frac{\sum_{\alpha \in S_{l,k}^{(1)}} p[a[i] = \alpha | r[i]]}{\sum_{\alpha \in S_{l,k}^{(0)}} p[a[i] = \alpha | r[i]]} \quad (7)$$

根据贝叶斯公式,并设传输的符号是均匀分布的,可推出:

$$LLR(b_{l,k}) = \lg \frac{\sum_{\alpha \in S_{l,k}^{(1)}} p[r[i] | a[i] = \alpha]}{\sum_{\alpha \in S_{l,k}^{(0)}} p[r[i] | a[i] = \alpha]} \quad (8)$$

又因为 $r(i)$ 的条件概率密度函数是复高斯过程,则有:

$$LLR(b_{l,k}) = \lg \frac{\sum_{\alpha \in S_{l,k}^{(1)}} \exp\{-\frac{1}{2} \frac{|r[i] - G_{ch}(i)\alpha|^2}{\sigma^2}\}}{\sum_{\alpha \in S_{l,k}^{(0)}} \exp\{-\frac{1}{2} \frac{|r[i] - G_{ch}(i)\alpha|^2}{\sigma^2}\}} \quad (9)$$

当然,在实际通信系统中,由于考虑到降低解调算法复杂度的需要,根据 $\ln \sum_i e^{x_i} \approx \max_i \{x_i\}$,这里要求 x_i 的均值较小,且 $r[i] = G_{ch}(i) \cdot y[i]$,经过化解和 $2/\sigma^2$ 规范化后,最终的软信息公式可以写作:

$$LLR(b_{l,k}) = \frac{|G_{ch}(i)|^2}{4} \{ \min_{\alpha \in S_{l,k}^{(0)}} |y[i] - \alpha|^2 - \min_{\alpha \in S_{l,k}^{(1)}} |y[i] - \alpha|^2 \} \quad (10)$$

$$\text{令 } S_{l,k} = \frac{\min_{\alpha \in S_{l,k}^{(0)}} |y[i] - \alpha|^2 - \min_{\alpha \in S_{l,k}^{(1)}} |y[i] - \alpha|^2}{4},$$

那么结果可以写为:

$$LLR(b_{I,k}) = |G_{ch}(i)|^2 \cdot S_{I,k} \quad (11)$$

图6是16QAM的星座图,由于I路、Q路的映射是完全相同的,因此本文只介绍I路。

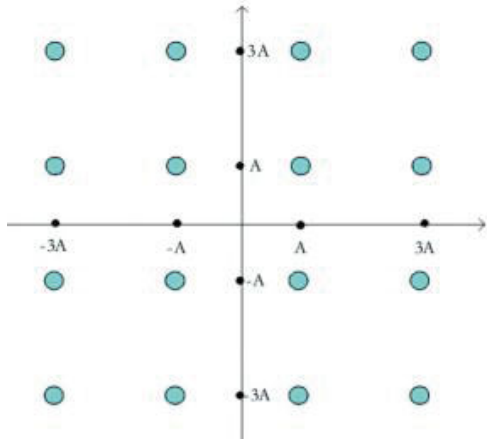


图6 16QAM星座图

根据公式(11),可以得到16QAM的 $S_{I,k}$ 的表达式为:

$$S_{I,1} = \begin{cases} y_I[i] & |y_I[i]| \leq 2 \\ 2(y_I[i]-1) & y_I[i] > 2 \\ 2(y_I[i]+1) & y_I[i] < -2 \end{cases} \quad (12)$$

$$S_{I,2} = -|y_I[i]| + 2 \quad (13)$$

对于Q路,只需要将 $S_{I,k}$ 改为 $S_{Q,k}$ 即可,这样就得到了整个16QAM的LLR表达式。

在程序的运行过程中,如果判断语句太多的话,会极大地浪费时间,故使用了符号函数sign()和绝对值函数abs()来去除判断语句,那么公式(12)和(13)可以表示为:

$$S_{I,1} = \{ [1 - \text{sign}(y_I[i] + 2)] * (y_I[i] + 3)^2 + [1 + \text{sign}(y_I[i] + 2)] * (y_I[i] + 1)^2 - [1 + \text{sign}(y_I[i] - 2)] * (y_I[i] - 3)^2 - [1 - \text{sign}(y_I[i] - 2)] * (y_I[i] - 1)^2 \} / 2 \quad (14)$$

$$S_{I,2} = -\text{abs}(y_I[i]) + 2 \quad (15)$$

在SSE指令中,没有直接计算sign()和abs()的指令,可使用以下步骤来求两个函数:

(1) 输入4个float型数据,设为 $(a1, a2, a3, a4)$,且 $a1 > 0, a2 < 0, a3 > 0, a4 < 0$;

(2) 每个数据乘以-1,得到 $(-a1, -a2, -a3, -a4)$;

(3) 对应位比较大小,取最大值得到 $(a1, -a2, a3, -a4)$ 即是原数据的绝对值;

(4) 第3步得到的结果再与输入数据对应位相除,就得到符号函数。

在软解调的过程中,使用_mm_load_ps指令去存储连续的两个符号(实部、虚部都是float型),下面对公式(14)和(15)进行SSE优化,伪代码如下:

```
m128 x=_mm_load_ps(symbols) //m128为声明
one=_mm_load_ps(1.0,1.0,1.0,1.0)
two=_mm_load_ps(2.0,2.0,2.0,2.0)
three=_mm_load_ps(3.0,3.0,3.0,3.0)
计算(y_I[i]+3)^2
m=_mm_add_ps(three,x); //(y_I[i]+3)
mm=_mm_mul_ps(m,m); //(y_I[i]+3)^2
计算[1-sign(y_I[i]+2)]
n=_mm_add_ps(two,x); //(y_I[i]+2)
n1=sign(n); //sign(y_I[i]+2)
n2=_mm_sub_ps(one,n1); //[1-sign(y_I[i]+2)]
那么[1-sign(y_I[i]+2)]*(y_I[i]+3)^2可以表示为:
P=_mm_mul_ps(n2,mm);
同理(14)的其他三部分也可以求出,这样的话,
整个S_{I,1}就可以表示出来。
S_{I,2}则为:
a=abs(x); //abs(y_I[i])
S_{I,2}=_mm_sub_ps(two,a);
```

4 结果比较

表1、2分别是FFT变换、软解调优化前后的时间对比。

表1 FFT变换优化前后的时间对比

点数	SSE优化后	SSE优化前
2048	14.776us	76.125us
1024	5.920554us	42.087us

表2 软解调优化前后的时间对比

	SSE优化后	SSE优化前
16QAM	1.561us	29.806us
64QAM	1.193us	9.134us

通过上面的数据分析可以看出,FFT变换和软解调SSE优化后的运行时间有大幅度的提高。

流水工作是把相邻的相关性很大的语句进行适当的分离,以便每条指令有充分的执行时间,如果没有这个工作,下一条指令必须等待上一条指令执行完毕后才能执行。它很好地避开了数据的相关性,保证了处理器工作的连续性,如图7所示:

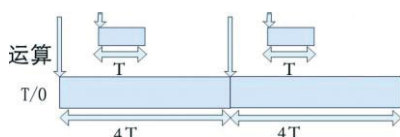


图7 流水工作示意图

在图7中,设每个数据存储的周期为 $4T$,SSE指令的计算时间为 T 。在数据进行存储的时候,可以进行其他数据的计算。

5 结论

本文研究了基于GPP-SDR的HSUPA接收机的软件设计,依托Windows操作系统环境对各个物理信道进行了C++语言编程实现,以及频域均衡中的FFT变换和软解调的SSE优化,使软件的运行效率有了很大的提高,能够满足TD-HSUPA系统实时通信的需求,同时也为C-RAN的后续研究和实现提供了重要的参考意义。

参考文献:

- [1] 卫国祥. 基于TD-SCDMA的HSUPA技术研究[C]. 第十届中国科协年会信息化与社会发展学术讨论会分会场论文集, 2008.
- [2] 雷杰,赵熠飞,黄敏,等. 基于GPP SDR的LTE系统RLC协议层实现研究[J]. 移动通信, 2011(02): 45-50.
- [3] 彭木根,王文博. TD-SCDMA移动通信系统[M]. 3版. 北京: 机械工业出版社, 2009.
- [4] 张晓琳. TD-HSUPA物理层算法仿真和实现[D]. 北京: 北京邮电大学, 2010.
- [5] 3GPP 25.221-921 Physical channels and mapping of

transport channels onto physical channels(TDD)(Release 9)[S].

- [6] 3GPP 25.222-920 Multiplexing and channel coding (TDD) (Release 9)[S].
- [7] 3GPP 25.223-900 Spreading and modulation (TDD) (Release 9)[S].
- [8] 孙远道,葛万成. 频域均衡在超三代移动通信系统中的应用研究[J]. 通信技术, 2010(06): 107-109.
- [9] 程佩青. 数字信号处理教程[M]. 3版. 北京: 清华大学出版社, 2007.
- [10] 顾昕钰,吴伟陵. 一种通用的高效软输出高阶解调算法[J]. 无线电工程, 2004(12): 16-20.

作者简介



王俊华: 厦门大学信息科学与技术学院硕士研究生, 研究方向为数字信号处理、TD-SCDMA物理层协议机制与实现。



齐心: 清华大学信息科学与技术国家实验室助理研究员, 主要研究领域包括多用户MIMO-OFDM系统中的信号处理和无线资源管理, 以及3GPP UMTS、LTE标准和实现。



黄联芬: 博士, 副教授, 现任职于厦门大学信息科学与技术学院通信工程系, 主讲《信号与系统》、《数字信号处理》和《现代信号处理》课程, 研究方向为无线通信技术、通信信号处理、个人通信系统与网络和无线通信抗干扰等。